

Aplikasi Sensor Vision untuk Deteksi MultiFace dan Menghitung Jumlah Orang

RD. Kusumanto¹, Wahyu S. Pambudi², Alan N. Tompunu¹

¹ Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya, Palembang 30139

Email : manto_6611@yahoo.co.id; alan_tompunu@mail.polisriwijaya.ac.id; alan_polsri@yahoo.com

² Jurusan Teknik Elektro, Universitas Internasional Batam, Batam 29442

Email : pambudi@uib.ac.id; yoe2fa@yahoo.com

ABSTRAK

Perkembangan teknologi yang terkait dengan pemrosesan menggunakan komputer demikian pesatnya, salah satunya adalah image (citra) atau disebut Pengolahan Citra Digital (Digital Image Processing). Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Secara matematis, citra merupakan fungsi kontinyu dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit.

Computer Vision merupakan cabang ilmu Pengolahan Citra Digital yang memungkinkan komputer dapat melihat seperti manusia, sehingga dapat mengambil keputusan, melakukan aksi, dan mengenali suatu Objek. Bentuk implementasi dari Computer Vision adalah Face Detection, dimana salah satu metodenya Haar-like Features. Metode ini menghitung perbedaan jumlah setiap piksel pada daerah persegi panjang (rectangular) yang berdekatan pada lokasi tertentu dalam jendela deteksi.

Berdasarkan pengujian yang dengan menggunakan citra statis maupun citra dinamis, didapatkan kesimpulan bahwa metode haar-like sangat cocok digunakan untuk face detection dengan obyek citra statis maupun citra dinamis. Dengan menggunakan indikasi dari face detection untuk citra statis maupun citra dinamis maka jumlah orang akan diketahui. Pada pengujian dengan citra statis error 28% terjadi pada saat gambar berisi 7 orang.

Kata Kunci : Citra Digital, Haar-like Features, Face Detection

1. PENDAHULUAN

Perkembangan teknologi yang terkait dengan pemrosesan menggunakan komputer demikian pesatnya, dimana pemrosesan image (citra) yang awalnya menggunakan teknologi analog mulai bergeser menggunakan teknologi digital. Disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra secara digital disebut sebagai Pengolahan citra digital (Digital Image Processing). Proses digital disini mempunyai maksud bahwa pengolahan citra dilakukan secara digital menggunakan komputer [1]. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Secara matematis, citra merupakan fungsi kontinyu dengan intensitas cahaya pada bidang dua dimensi. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris.

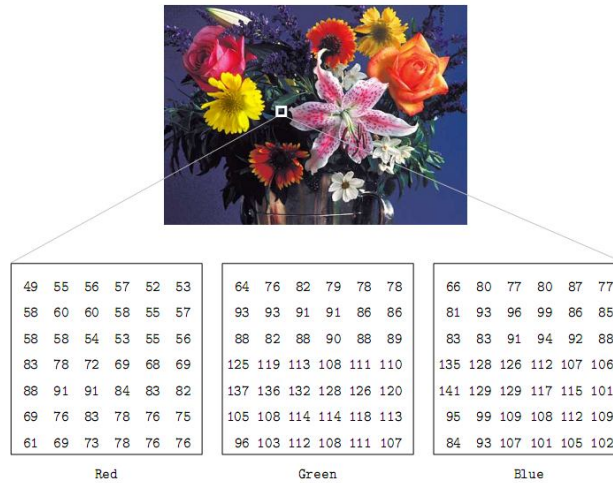
EmguCV adalah sebuah API (Application Programming Interface) library yang sudah sangat familiar pada Pengolahan Citra Digital untuk Computer Vision. Computer Vision itu sendiri merupakan cabang dari bidang ilmu Pengolahan Citra Digital yang memungkinkan komputer dapat melihat seperti manusia [2], sehingga dapat mengambil keputusan, melakukan aksi, dan mengenali suatu Objek. Bentuk implementasi dari Computer Vision adalah Face Detection [3], dimana teknik yang digunakan ada beberapa metode [4]. Haar-like Features merupakan salah satu metode yang digunakan untuk real time face detection, metode ini pertama kali dikenalkan oleh Viola Jones dan dikembangkan oleh Lienhart [5,6]. Metode ini menghitung perbedaan jumlah setiap piksel pada daerah persegi panjang (rectangular) yang berdekatan pada lokasi tertentu dalam jendela deteksi.

2. PENGOLAHAN CITRA DIGITAL.

Pengolahan citra digital (Digital Image Processing) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Pada aplikasi pengolahan citra digital pada umumnya, citra digital dapat dibagi menjadi 3, color image, black and white image dan binary image.

1. Color Image atau RGB (Red, Green, Blue).

Pada color image ini masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (Red), hijau (Green) dan biru (Blue). Jika masing-masing warna memiliki range 0 - 255, maka totalnya adalah $255^3 = 16.581.375$ (16 K) variasi warna berbeda pada gambar, dimana variasi warna ini cukup untuk gambar apapun. Karena jumlah bit yang diperlukan untuk setiap pixel, gambar tersebut juga disebut gambar-bit warna. Color image ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau dan biru untuk setiap pikselnya, seperti yang ditunjukkan gambar 1.



Gambar 1: *Color Image*[7].

2. *Black and White.*

Citra digital *black and white* (*grayscale*) setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Rentang tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit, atau 1 byte. Rentang warna pada *black and white* sangat cocok digunakan untuk pengolahan file gambar. Salah satu bentuk fungsinya digunakan dalam kedokteran (X-ray). *Black and white* sebenarnya merupakan hasil rata-rata dari *color image*, dengan demikian maka persamaannya dapat dituliskan sebagai berikut :

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \dots\dots\dots (1)$$

dimana $I_R(x, y)$ = nilai piksel *Red* titik (x, y) , $I_G(x, y)$ = nilai piksel *Green* titik (x, y) , $I_B(x, y)$ = nilai piksel *Blue* titik (x, y) sedangkan $I_{BW}(x, y)$ = nilai piksel *black and white* titik (x, y) .



Gambar 2: *Black and White (Grayscale)*[7]

3. *Binary Image.*

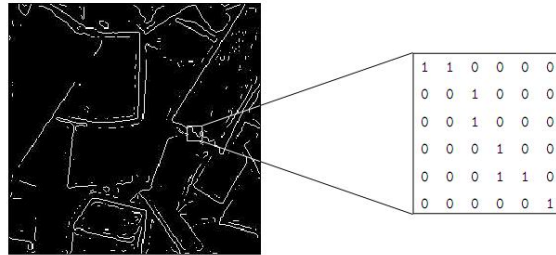
Setiap piksel hanya terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk setiap piksel, maka hanya perlu 1 bit per piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), sehingga sangat efisien dalam hal penyimpanan. Gambar yang direpresentasikan dengan biner sangat cocok untuk teks (dicetak atau tulisan tangan), sidik jari (*finger print*), atau gambar arsitektur. *Binary image* merupakan hasil pengolahan dari *black and white image*, dengan menggunakan fungsi sebagai berikut :

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 255 & I_{BW}(x, y) \geq T \end{cases} \dots\dots\dots (2)$$

dan dalam bentuk *floating point*

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 1 & I_{BW}(x, y) \geq T \end{cases} \dots\dots\dots (3)$$

dimana $I_{BW}(x, y)$ = nilai piksel *Gray* titik (x, y) , $I_{Bin}(x, y)$ = nilai piksel *Binary* titik (x, y) , sedangkan T adalah nilai *threshold*.



Gambar 3: *Binary Image*[7]

3. COMPUTER VISION

Terminologi lain yang berkaitan erat dengan pengolahan citra digital adalah *computer vision* atau *machine computer*. Pada hakikatnya, *computer vision* mencoba meniru cara kerja visual manusia (*Human Vision*). *Human Vision* sesungguhnya sangat kompleks, manusia melihat obyek dengan indera penglihatan (mata) lalu obyek citra diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti obyek apa yang tampak dalam pandangan matanya. Hasil interpretasi ini mungkin digunakan untuk mengambil suatu keputusan. Sebagaimana layaknya mata dan otak, *computer vision* adalah suatu sistem yang mempunyai kemampuan untuk menganalisis obyek secara visual, setelah data obyek yang bersangkutan dimasukkan dalam bentuk citra [8,9,10,11].

Proses-proses dalam *computer vision* dibagi dalam 3 (tiga) aktifitas :

1. Memperoleh atau mengakuisisi citra digital, proses ini bisa disebut juga sebagai proses *image preprocessing*.
2. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra berupa piksel.
3. Menganalisis dan menginterpretasi citra menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau manufaktur dan lain-lain [10]

Pengolahan citra merupakan proses awal (*preprocessing*) pada *computer vision*, sedangkan pengenalan pola merupakan proses untuk menginterpretasi citra. Teknik-teknik di dalam pengenalan pola memainkan peranan penting dalam *computer vision* untuk mengenali objek.

4. SENSOR VISION

Bentuk dari sensor vision adalah kamera digital atau webcam. Sensor vision adalah sensor yang paling kompleks digunakan dalam robotika. Sensor ini tidak banyak digunakan pada *embedded system* sampai saat ini, karena keterbatasan kecepatan prosesor dan kapasitas memori yang diperlukan untuk pengolahannya.



Gambar 2.7 *Sensor vision (Webcam)* [12]

Dalam beberapa tahun terakhir sensor vision yang berupa webcam telah mengalami pergeseran dalam teknologi. Teknologi sebelumnya menggunakan chip sensor CCD (charge coupled device) sedangkan saat menggunakan chip sensor

CMOS (complementary metal oxide semiconductor). Sensitivitas rentang kecerahan sensor CMOS lebih besar beberapa kali lipat dari sensor CCD. Apabila digunakan untuk *interface* dengan *embedded system* teknologi ini tidak ada perbedaan.

5. DETEKSI OBYEK DENGAN HAAR-LIKE.

Penelitian mengenai deteksi dan pengenalan wajah teknologi computer vision telah banyak dilakukan, salah satunya adalah menggunakan Haar like feature yang dikenal sebagai Haar Cascade Classifier. Haar-like features merupakan rectangular (persegi) features, yang memberikan indikasi secara spesifik pada sebuah gambar atau *image*. Ide dari Haar-like features adalah untuk mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari *image* obyek tersebut. Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah *image*[5]. Metode ini merupakan metode yang menggunakan *statistical model (classifier)*. Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat konsep utama :

1. Training data
2. Fitur segi empat sederhana yang disebut fitur Haar.
3. Integral image untuk pendeteksian fitur secara cepat.
4. Pengklasifikasi bertingkat (Cascade classifier) untuk menghubungkan banyak fitur secara efisien.

5.1 Training data pada Haar

Metode ini memerlukan 2 tipe gambar objek dalam proses *training* yang dilakukan, yaitu :

1. *Positive samples*, Berisi gambar obyek yang ingin di deteksi, apabila ingin mendeteksi mata maka *positive samples* ini berisi gambar wajah, begitu juga obyek lain yang ingin dikenali.
2. *Negative samples*, Berisi gambar obyek selain obyek yang ingin dikenali, umumnya berupa gambar *background* (tembok, pemandangan, lantai, dan gambar lainnya). Resolusi untuk sampel negatif disarankan untuk memiliki resolusi yang sama dengan resolusi kamera.

Training dari Haar menggunakan dua tipe sampel diatas. Informasi dari hasil training ini lalu dikonversi menjadi sebuah parameter model statistik.

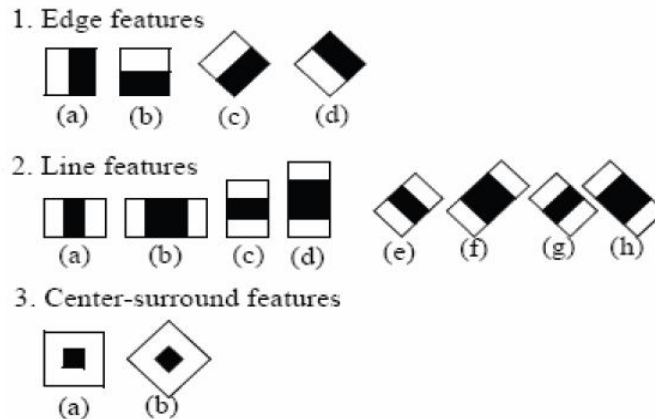
5.2 Sistem Kerja Algoritma Haar Cascade Classifier

Algoritma Haar menggunakan metode statistical dalam melakukan pendeteksian wajah. Metode ini menggunakan sample *haarlike fetures*. Classifier ini menggunakan gambar berukuran tetap (umumnya berukuran 24x24). Cara kerja dari haar dalam mendeteksi wajah adalah dengan menggunakan teknik *sliding window* berukuran 24x24 pada keseluruhan gambar dan mencari apakah terdapat bagian dari gambar yang berbentuk seperti wajah atau tidak. Haar juga memiliki kemampuan untuk melakukan scaling sehingga dapat mendeteksi adanya wajah yang berukuran lebih besar ataupun lebih kecil dari gambar pada classifier [6].

Tiap *feature* dari *haar-like feature* didefinisikan pada bentuk dari *feature*, diantaranya koordinat dari *feature* dan juga ukuran dari *feature* tersebut.

5.3 Haar Feature

Haar Feature adalah fitur yang didasarkan pada *Wavelet Haar* [5]. *Wavelet Haar* adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah). Untuk dua dimensi, satu terang dan satu gelap. Selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik. Setiap *Haar-like feature* terdiri dari gabungan kotak - kotak hitam dan putih.



Gambar 4. Macam-macam variasi feature pada Haar [6]

3 tipe kotak(rectangular) feature:

1. Tipe *two-rectangle feature* (horizontal/vertikal)
2. Tipe *three-rectangle feature*
3. Tipe *four-rectangle feature*

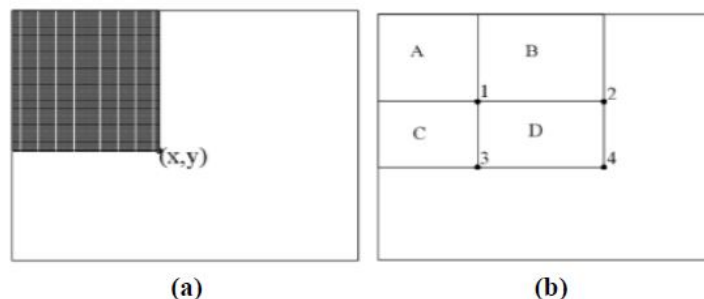
Adanya fitur Haar ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang. Jika nilai perbedaannya itu diatas nilai ambang atau *threshold*, maka dapat dikatakan bahwa fitur tersebut ada. Nilai dari *Haar-like feature* adalah perbedaan antara jumlah nilai-nilai piksel *gray level* dalam daerah kotak hitam dan daerah kotak putih:

$$f(x) = \text{SumBlack rectangle} - \text{SumWhite rectangle} \dots\dots\dots (4)$$

dimana untuk kotak pada Haar-like feature dapat dihitung secara cepat menggunakan “*integral image*” .

5.4 Integral Image

Integral Image digunakan untuk menentukan ada atau tidaknya dari ratusan fitur Haar pada sebuah gambar dan pada skala yang berbeda secara efisien. Pada umumnya, pengintegrasian tersebut berarti menambahkan unit-unit kecil secara bersamaan. Dalam hal ini unit-unit kecil tersebut adalah nilai-nilai piksel. Nilai integral untuk masing-masing piksel adalah jumlah dari semua piksel-piksel dari atas sampai bawah. Dimulai dari kiri atas sampai kanan bawah, keseluruhan gambar itu dapat dijumlahkan dengan beberapa operasi bilangan bulat per piksel.



Gambar 5. *Integral Image* [5]

Seperti yang ditunjukkan oleh gambar 5(a) di atas setelah pengintegrasian, nilai pada lokasi piksel (x,y) berisi jumlah dari semua piksel di dalam daerah segiempat dari kiri atas sampai pada lokasi (x,y) atau daerah yang diarsir. Guna mendapatkan nilai rata-rata piksel pada area segiempat (daerah yang diarsir) ini dapat dilakukan hanya dengan membagi nilai pada (x,y) oleh area segiempat.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \dots\dots\dots (5)$$

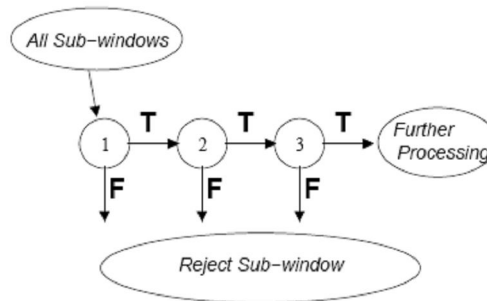
dimana $ii(x, y)$ adalah *integral image* dan $i(x, y)$ adalah *original image*

Guna mengetahui nilai piksel untuk beberapa segiempat yang lain, seperti segiempat D pada gambar 5(b), dapat dilakukan dengan cara menggabungkan jumlah piksel pada area segiempat A+B+C+D, dikurangi jumlah dalam segiempat A+B dan A+C, ditambah jumlah piksel di dalam A. Dengan, A+B+C+D adalah nilai dari integral image pada lokasi 4, A+B adalah nilai pada lokasi 2, A+C adalah nilai pada lokasi 3, dan A pada lokasi 1. Sehingga hasil dari D dapat dikomputasikan.

$$D = (A + B + C + D) - (A + B) - (A + C) + A \dots\dots\dots (6)$$

5.5 Cascade Classifier

Cascade classifier adalah sebuah rantai *stage classifier*, dimana setiap *stage classifier* digunakan untuk mendeteksi apakah didalam image *sub window* terdapat obyek yang diinginkan (*object of interest*). Stage classifier dibangun dengan menggunakan algoritma adaptive-boost (*AdaBoost*). Algoritma tersebut mengkombinasikan performance banyak weak classifier untuk menghasilkan *strong classifier*. *Weak classifier* dalam hal ini adalah nilai dari haar-like feature. Jenis *AdaBoost* yang digunakan adalah *Gentle AdaBoost*.



Gambar 6. Model *classifier* secara *cascade* [5].

6. APLIKASI DENGAN EMGUCV

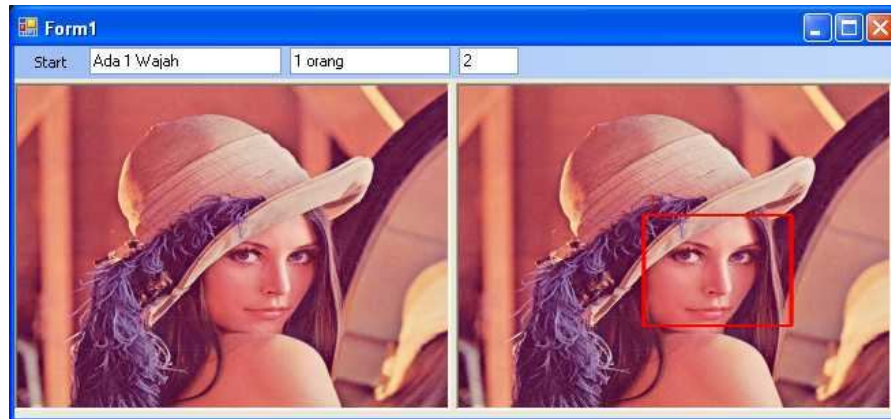
EmguCV adalah cross platform yang terdapat dalam .NET untuk *library* pengolahan citra pada Intel OpenCV. EmguCV ini mengikuti fungsi yang terdapat pada OpenCV yang diambil dari .NET oleh sebab itu *compatible* dengan bahasa pemrograman C#, VB, VC++, IronPython dan sebagainya. Program ini bersifat *opensource* sehingga sangat cocok apabila digunakan untuk penelitian, salah satunya adalah untuk aplikasi *computer vision*.

7. HASIL DAN PEMBAHASAN.

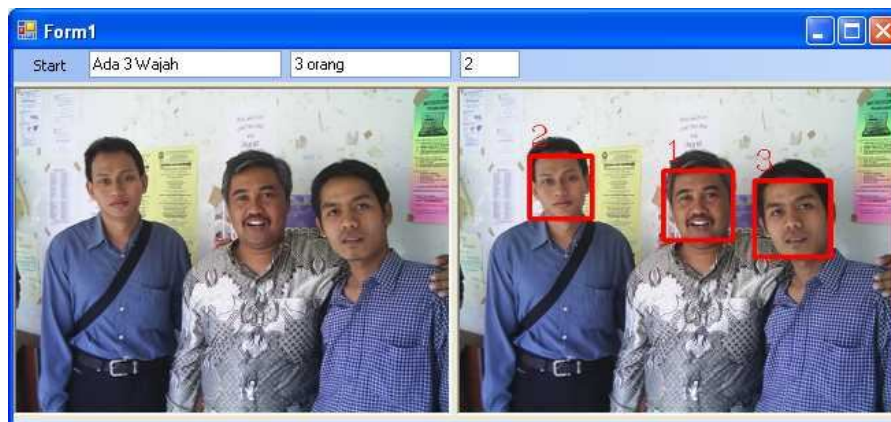
Pada penelitian ini obyek citra yang akan digunakan adalah citra statis dan citra dinamis. Citra statis ini adalah berupa file *image* dimana dalam gambar tersebut terdapat orang, sedangkan citra dinamis adalah *image* yang bersumber dari sensor vision atau *webcam*.

7.1 Pengujian dengan Citra Statis.

Langkah awal untuk melakukan pengujian dengan citra statis ini adalah mendapatkan file image dimana dalam gambar tersebut terdapat orang lebih dari 1. Apabila sudah didapatkan maka file tersebut diolah dengan program yang telah dibuat sebelumnya. Tujuan dari pengujian ini adalah sebagai pengujian awal sebelum ke proses yang menggunakan citra dinamis.



Gambar 7. Citra Statis dengan 1 Orang.



Gambar 8. Citra Statis dengan 3 Orang.

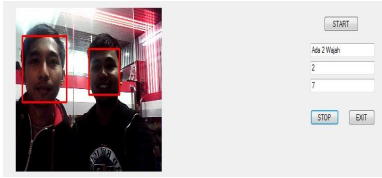


Gambar 9. Citra Statis dengan 7 Orang.

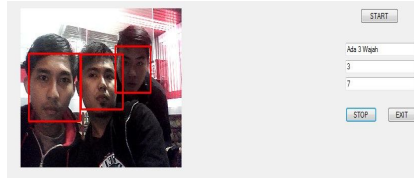
Berdasarkan hasil pengujian dengan citra statis pada gambar 7,8 dan 9, dapat disimpulkan bahwa dengan menggunakan metode *haar-like* untuk *face detection* dapat digunakan untuk menghitung jumlah orang. *Error* yang terjadi pada saat pengujian dengan menggunakan gambar 7 orang, dimana yang seharusnya 7 dideteksi 9 orang atau bisa dikatakan memiliki *error* 28%.

7.2 Pengujian dengan Citra Dinamis.

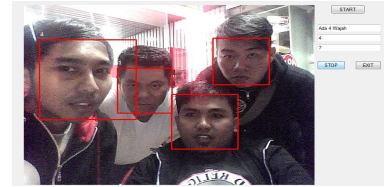
Pada pengujian ini, citra yang digunakan adalah dinamis, dimana citra berasal dari sensor vision berupa *webcam*. Hasil pengujian ini akan ditampilkan dalam gambar dibawah ini.



Gambar 10. Citra Dinamis dengan 2 Orang.



Gambar 11. Citra Dinamis dengan 3 Orang.



Gambar 12. Citra Dinamis dengan 4 Orang.

Berdasarkan hasil pengujian dengan citra dinamis pada gambar 10, 11 dan 12, dapat disimpulkan bahwa dengan menggunakan metode *haar-like* untuk *face detection* dapat digunakan juga untuk menghitung jumlah orang secara *real time* yang bersumber pada citra dari sensor vision.

8. PENUTUP

8.1 Kesimpulan.

Berdasarkan pengujian dengan menggunakan citra statis maupun citra dinamis, didapatkan kesimpulan sebagai berikut :

1. Metode *haar-like* sangat cocok digunakan untuk *face detection* dengan obyek citra statis maupun citra dinamis.
2. Berdasarkan indikasi dari *face detection* untuk citra statis maupun citra dinamis maka jumlah orang akan diketahui.
3. Pada pengujian dengan citra statis *error* 28% pada saat gambar berisi 7 orang.

8.2 Rekomendasi.

Berdasarkan penelitian yang dilakukan, penerapan dari *haar-like* ini apabila dikombinasikan dengan metode untuk *face Recognition* akan dapat digunakan sebagai alternative sistem absensi. Penggunaan gabungan dua metode ini akan mampu memberikan performa terbaik sebagai model sistem absensi yang canggih.

DAFTAR PUSTAKA

- [1] Sutoyo. T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto, “*Teori Pengolahan Citra Digital*”, Andi Yogyakarta dan UDINUS Semarang, 2009.
- [2] Y. Amit, D. Geman, and K. Wilder. “*Joint Induction of Shape Features and Tree Classifiers*”, 1997.
- [3] Bradski Gary , Kaehler Adrian,”*Learning OpenCV*”, O’Reilly Media, 1005 Gravenstein Highway North, Sebastopol, CA 95472, September 2008.
- [4] Schneiderman, H. and T. Kanade (2000).” *A statistical method for 3D object detection applied to faces and cars*”. In International Conference on Computer Vision.
- [5] Viola, Paul and Michael Jones (2001). “*Rapid object detection using boosted cascade of simple features*”. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, 2001.
- [6] Lienhart, Rainer and Jochen Maydt (2002). “*An extended set of haar-like features for rapid object detection*”. In: IEEE ICIP 2002, Vol.1, pp 900-903.
- [7] McAndrew Alasdair, (2004), *An Introduction to Digital Image Processing with Matlab. Notes for SCM2511 Image Processing I*, School of Computer Science and Mathematics Victoria University of Technology.
- [8] Wahyu Setyo Pambudi, Irma Salamah, Alan Novi Tomponu, (2011), *Deteksi dan Estimasi Jarak Obyek Menggunakan Single Camera Dengan Model Segmentasi HSV*, Seminar Nasional Teknoin 2011.
- [9] RD. Kusumanto, Alan Novi Tomponu, Wahyu Setyo Pambudi, “*Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV*”, Seminar Nasional Teknik Elektro (SNTE) 2011, Politeknik Negeri Jakarta, 8 September 2011.
- [10] Wahyu Setyo Pambudi, “*Rancang Bangun 3 Wheels Omni-directional Mobile Robot Menggunakan Sensor Position Sensitive Device (PSD) serta Sensor Vision dengan Metode Kendali Fuzzy Logic Controller (FLC) untuk Menghindari Halangan*”, Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011), UDINUS, 16 April 2011.
- [11] RD. Kusumanto, Alan Novi Tomponu, “*Pengolahan Citra Digital untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB*”, Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011), UDINUS, 16 April 2011.
- [12] <http://www.citrajaya.net/produk/genius/850-genius-webcam-islam-310.html>, 2011.